

# Ein Framework für kontextbezogene Anwendungen in der Nexus-Plattform

Matthias Wieland

Daniela Nicklas

Institut für Architektur von Anwendungssystemen  
Universität Stuttgart  
Universitätsstraße 38, 70569 Stuttgart  
wielanms@informatik.uni-stuttgart.de

Institut für Parallele und Verteilte Systeme  
Universität Stuttgart  
Universitätsstraße 38, 70569 Stuttgart  
danickla@informatik.uni-stuttgart.de

**Abstract:** Im Rahmen des Nexus-Projekts wurden zahlreiche kontextbezogene Anwendungen entwickelt. Aus diesen Erfahrungen heraus wurde ein Framework erarbeitet, das die Entwicklung orts- und kontextbezogener Anwendungen durch geeignete Abstraktionen und Laufzeitbibliotheken unterstützt. Dieser Beitrag beschreibt den Aufbau und die Vorteile dieses Frameworks. Zum Schluss geht der Beitrag noch auf die zukünftige Entwicklung des Nexus-Projekts im Bereich Anwendungsunterstützung ein.

## 1 Einleitung

Die Entwicklung von kontextbezogenen Anwendungen ist sehr teuer, fehleranfällig und kompliziert, wenn keine Techniken eingesetzt werden, um die Wiederverwendung von Kontextdaten und Software-Komponenten zu ermöglichen. Die Nexus-Plattform [SFB627] ist eine Infrastruktur, welche eine Entwicklung solcher Anwendungen durch ein umfassendes Umgebungsmodell unterstützt [Ni05], durch welches Angebote verschiedener Datenabbieter dynamisch föderiert werden. Dadurch wird die Wiederverwendung der Umgebungsmodelldaten ermöglicht. Zudem werden von der Nexus-Plattform zahlreiche Dienste angeboten, wie z. B. der Ereignisdienst, Geonachrichtendienst oder Kartendienst.

Um die Funktion der Plattform zu demonstrieren, wurden zahlreiche Nexus-Anwendungen entwickelt [NGS03, NHM+04]. Diese Anwendungen wurden bisher immer von Grund auf neu entwickelt. Um diesen Entwicklungsprozess zu verbessern, wurde ein Framework für die Entwicklung kontextbezogener Anwendungen geschaffen [Wi05]. Dies soll zur Kostenreduzierung, Vereinfachung und Standardisierung bei der Nexus-Anwendungsentwicklung führen. Erreicht wurden diese Ziele durch die Anwendung der softwaretechnischen Grundkonzepte eines Frameworks, -Modularität, Wiederverwendbarkeit, Erweiterbarkeit und Umkehr des Kontrollflusses.

Es gibt diverse Frameworks, die zur Entwicklung kontextbezogener Anwendungen eingesetzt werden können [HeIn04, DeAb00, DSA01]. Allerdings sind diese nicht auf den Einsatz im Zusammenspiel mit der Nexus-Plattform ausgelegt.

Im Folgenden wird der Aufbau und die Grundkonzepte des entwickelten Frameworks beschrieben. Danach wird ein Ausblick auf die zukünftigen Vorhaben im Rahmen der Anwendungsunterstützung im Nexus-Projekt gegeben.

## 2 Aufbau des Frameworks

Das Framework besteht aus drei unterschiedlichen Teil-Frameworks [Wi05]. Diese drei Frameworks können auch getrennt voneinander eingesetzt werden. Das **Entwicklungs-Framework** besteht aus den Wrappern zum Zugriff auf die Nexus-Bibliotheken. Es wird über feste Schnittstellen vom Anwendungsprogrammierer genutzt und wird in den Sourcecode einer Nexus-Anwendung eingebunden. Alle Funktionen dieses Frameworks sind bereits zur

Compile-Zeit bekannt und fest verknüpft. Die Bibliotheken bieten allgemeine Funktionen zur Erhebung von Kontextinformationen und für die Abfrage von Kontextinformationen aus dem Umgebungsmodell.

Der zweite Baustein des Frameworks ist das **Laufzeit-Framework**. Es hat die Aufgabe, die Endbenutzer der Anwendungen zu unterstützen. Das Laufzeit-Framework ist eine eigenständige Anwendung, die auf dem Endgerät abläuft. Sie wird vom Benutzer des Endgerätes verwendet, um z. B. Nexus-Anwendungen zu finden und diese einfach zu installieren, zu konfigurieren und schließlich auszuführen. Es bietet jedoch auch Nexus-Anwendungen Unterstützung. Zum Beispiel kann das Laufzeit-Framework die Anwendungen über ihren aktuellen Ausführungskontext informieren. Außerdem verwaltet es beschränkte Ressourcen, wie z. B. Sensoren des Endgerätes, die sonst immer nur von einer Anwendung gleichzeitig genutzt werden könnten. Diese Funktionen können nur zur Laufzeit angeboten werden und sind darum nicht im Entwicklungs-Framework sondern im Laufzeit-Framework enthalten.

Das **Projekt-Framework** unterstützt den Entwicklungsprozess und somit den Entwickler mit Werkzeugen. Dieser Baustein wurde noch nicht implementiert. Einige Ideen dieses Framework-Teils werden in der zukünftigen Forschung näher untersucht. Eine kleine Auswahl an Themen wird in Kapitel 5 vorgestellt.

### 3 Aufgabenbereiche einer kontextbasierten Anwendung

In Abbildung 1 werden die konzeptionellen Schnittstellen einer kontextbasierten Nexus-Anwendung dargestellt. Die Anwendung kommuniziert über das Nexus-Interface mit der Nexus-Plattform, um Informationen über das Umgebungsmodell zu erhalten. Über die grafische Benutzerschnittstelle (User Interface) interagiert der Benutzer mit der Anwendung. Des Weiteren greift sie auf lokal am Endgerät angeschlossene Sensoren zu, um ihren Ausführungskontext zu ermitteln (Local System Interface). Außerdem nutzt sie entfernte Dienste, die Änderungen in der realen Umgebung bewirken (Remote Services Interface). Durch Framework-Unterstützung in diesen Hauptaufgabenbereichen kann die Nexus-Anwendungsentwicklung vereinfacht werden. Das entwickelte Framework bietet jedoch nur Unterstützung im Bereich des Nexus Interface und des Local System Interface. Die anderen Schnittstellen sind Gegenstand der Forschung und werden zukünftig an Bedeutung gewinnen.

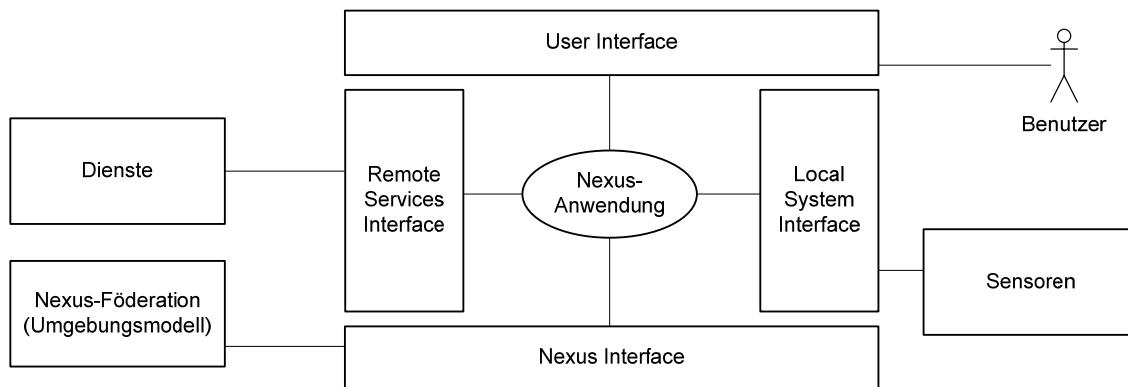


Abbildung 1: Die Schnittstellen einer Nexus-Anwendung

### 4 Konzepte des Frameworks

In diesem Kapitel werden die Konzepte vorgestellt, die dem Entwurf und der Implementierung des Frameworks zugrunde liegen.

**Komponentenbasiertes System:** Das Framework ist aus Komponenten, die festgelegte

Schnittstellen besitzen, aufgebaut. Diese Komponenten können flexibel miteinander verbunden werden. Das bedeutet aus Sicht der Anwendungen, dass die Komponenten entweder extern oder intern genutzt werden können. Werden die Komponenten extern genutzt, werden sie als eigenständige Prozesse gestartet und kommunizieren mittels Interprozesskommunikation. Intern genutzt bedeutet, sie werden fest in die Anwendungen eingebunden und sind somit Teil der Anwendung. Dadurch können sie direkt aufgerufen werden und laufen im Prozess der Anwendungen ab.

**Robuste lokale Kontexterfassung:** Die Erfassung der Kontextinformationen ist für Nexus-Anwendungen eine zentrale Aufgabe, darum soll sie zu einem möglichst großen Teil vom Framework übernommen werden. Das Kontexterfassungssystem des Frameworks ist: Robust gegen Ausfälle, unabhängig von der Sensor-Hardware, die genutzt wird, bietet Datenfusion bei mehrfach verfügbaren Sensordaten.

**Ereignisbasiertes System:** Um eine Umkehr des Kontrollflusses zu erreichen, wird an vielen Stellen im Framework ein ereignisbezogenes Benachrichtigungssystem eingesetzt. Dieses System besteht aus den folgenden Bestandteilen:

**Verteiltes System:** Um eine hohe Flexibilität des Frameworks zu erreichen, sollen dessen Komponenten als verteiltes System aufgebaut werden. Da die Subsysteme in eigenen Prozessen ablaufen, muss eine Technik für Interprozesskommunikation eingesetzt werden, um einen Informationsaustausch zwischen den Subsystemen zu ermöglichen. Für das Framework wurde Remote Method Invocation (RMI) gewählt.

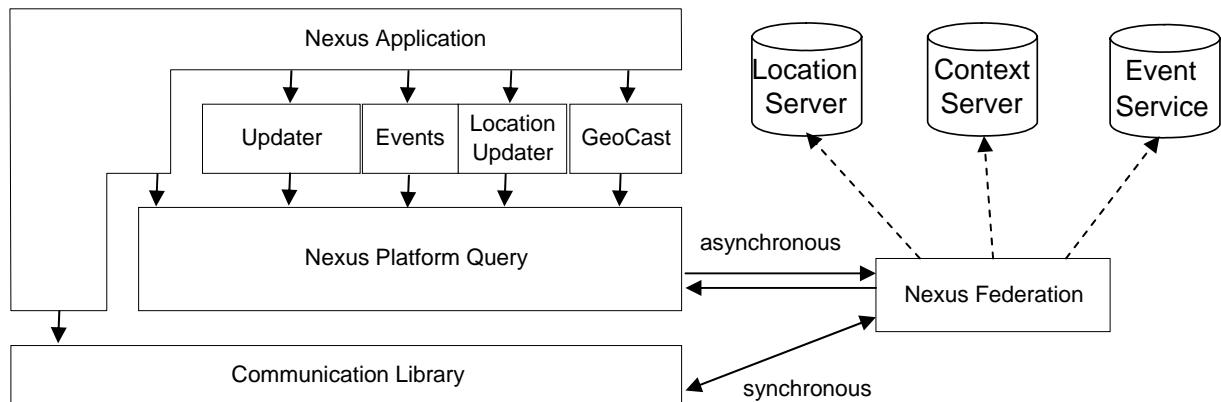


Abbildung 2: Die modulare Kommunikationsarchitektur des Frameworks

**Modulare Kommunikationsarchitektur:** Nexus-Anwendungen kommunizieren auf viele verschiedene Arten mit ihrer Umgebung. Damit sie immer auf dem gerade benötigten Abstraktionslevel Unterstützung finden, wird von dem Framework eine modulare Kommunikationsarchitektur, die aus drei Schichten besteht, bereitgestellt (siehe Abbildung 2). Die Anwendungen können je nach Wunsch mittels jeder dieser Schichten kommunizieren. Die Anwendungen werden dadurch in die Lage versetzt, mit der Nexus-Plattform und externen Diensten zu kommunizieren. Die unterste Schicht kapselt die zur Kommunikation eingesetzte Methode (z. B. RMI oder SOAP). Diese Schicht arbeitet synchron, d. h., die Anwendung wird bei der Nutzung blockiert, bis die Antwort des Servers verfügbar ist oder ein Timeout stattfindet. Die mittlere Schicht realisiert asynchrone, ereignisbasierte Kommunikationsfunktionen. Für jeden Nexus-Dienst existieren Stellvertreterobjekte, die mit diesen Kommunikationsfunktionen arbeiten. Schließlich stellt die oberste Schicht erweiterte Funktionen zur Verfügung, die auf den Basisfunktionen aufsetzen. Dies sind Funktionen, die im Anwendungsbereich oft benötigt werden und dafür optimierte Schnittstellen anbieten. Eine Funktion "Location Updater" übernimmt z. B. das Aktualisieren der eigenen Position im Location Server, ohne dass dafür extra eine Kontextanfrage erzeugt werden muss.

## 5 Zusammenfassung und Ausblick

Die Funktionalität des Frameworks wurde durch Neuentwicklung zweier ortsbezogener Beispieldaten überprüft. Dies hat gezeigt, dass die Anwendungsentwicklung durch das Framework erleichtert wird, und die Kosten reduziert werden. Bei bestehenden Anwendungen wurde nachgewiesen, dass durch den Einsatz des Frameworks die Qualität der Software verbessert werden kann.

Insgesamt zeigte sich durch die Erfahrungen im Nexus-Projekt, dass für die Entwicklung mobiler, orts- oder kontextbezogener Anwendungen neue oder angepasste Methoden, Werkzeuge und Entwurfsmuster notwendig sind, die durch den Stand der Technik nur unzureichend unterstützt werden [BDK+06]. Deswegen wird dieses Forschungsgebiet im Projekt Nexus in Zukunft verstärkt aus verschiedenen Blickwinkeln bearbeitet: Für große, prozessorientierte Anwendungen, wie sie im Umfeld der Smart Factory [BJR+03] auftreten, wird erforscht, wie der Kontrollfluss der Anwendungen mit Hilfe von kontextbasierten Workflows dargestellt werden kann. Dies wird sich vor allem für Anwendungen in prozessorientierten Umgebungen auszahlen [WLJ+06]. Des Weiteren wird für die Entwicklung verteilter kontextbasierter Anwendungen Werkzeugunterstützung entwickelt, ebenso für das Testen und Debuggen, da sich dies wesentlich vom Testen normaler Anwendungen unterscheidet. Im Bereich der Kontextverarbeitung wird eine weitere Abstraktionsebene eingeführt. Aus Basiskontext wird höherwertiger Kontext, so genannte Situationen, abgeleitet. Diese Situationen stehen allen Anwendungen zur Verfügung, werden aber von der Plattform erkannt und überwacht.

## Literaturverzeichnis

- [NGS03] D. Nicklas; M. Grossmann; T. Schwarz: NexusScout: An Advanced Location-Based Application On A Distributed, Open Mediation Platform, Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003
- [NHM+04] D. Nicklas; N. Höhne; M. Moltenbrey; B. Mitschang: Design and Implementation Issues for Explorative Location-based Applications: the NexusRallye, Proceedings for the VI Brazilian Symposium on GeoInformatics: GeoInfo 2004
- [HeIn04] K. Henricksen; J. Indulska: A Software Engineering Framework for Context-Aware Pervasive Computing, Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04), Orlando, Florida, March 14 - 17, 2004, pp. 77-86
- [DeAb00] A. Dey; G. Abowd: The Context Toolkit: Aiding the Development of Context-Aware Applications, Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6, 2000
- [DSA01] A. Dey; D. Salber; G. Abowd: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human-Computer Interaction, 16. 2001
- [SFB627] Sonderforschungsbereich 627: Nexus, <http://www.nexus.uni-stuttgart.de/>
- [Wi05] M. Wieland: Entwicklung eines Frameworks für Nexus-Anwendungen (Diplomarbeit), 2005
- [BJR+03] M. Bauer; L. Jendoubi; K. Rothermel; E. Westkämper: Grundlagen ubiquitärer Systeme und deren Anwendung in der "Smart Factory". In: Gronau, Norbert (ed.); Krallmann, Hermann; Scholz-Reiter, Bernd (ed.): Industrie Management - Zeitschrift für industrielle Geschäftsprozesse. Bd. 19(6), 2003
- [WLJ+06] M. Wieland; F. Leymann; L. Jendoubi; D. Nicklas; F. Dürr: „Task-orientierte Anwendungen in einer Smart Factory“, 1. Konferenz Mobilität und Mobile Informationssysteme (MMS 2006), Lecture Notes in Informatics (LNI), P-76, 2006
- [Ni05] D. Nicklas: „Ein umfassendes Umgebungsmodell als Integrationsstrategie für ortsbezogene Daten und Dienste“, Dissertation, Universität Stuttgart, 2005
- [BDK+06] C. Becker; F. Dürr; M. Knoll; D. Nicklas; T. Weis: „Entwicklung ortsbezogener Anwendungen.“ Praxis der Informationsverarbeitung und Kommunikation. (angenommen), 2006